

PATENT ABSTRACTS OF JAPAN

(11) Publication number : 11-184639

(43) Date of publication of application : 09.07.1999

(51) Int.CI.

G06F 3/06

(21) Application number : 09-350510

(71) Applicant : HITACHI LTD

(22) Date of filing : 19.12.1997

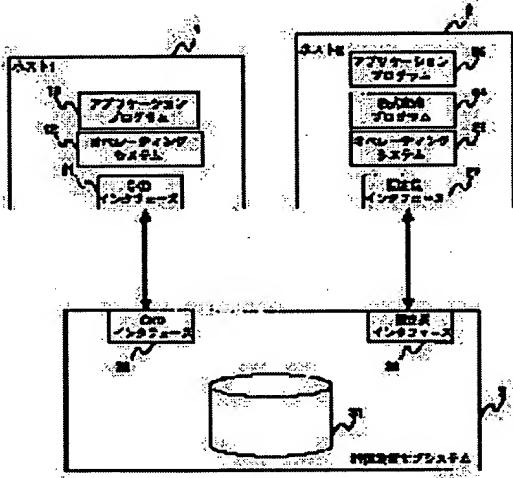
(72) Inventor : KITAMURA MANABU
YAMAMOTO AKIRA
HONMA SHIGEO

(54) COMPUTER SYSTEM

(57) Abstract:

PROBLEM TO BE SOLVED: To make data possible to be shared between systems different in format of data to be stored in storage devices.

SOLUTION: A disk 31 is a disk device according to a fixed length format and a storage device subsystem 3 stores records in CKD format coming from a host 1 while shaping them for the unit of a fixed length. In a format transformation program 24, the fixed length block of the disk 31 is read out, the data of the CKD format stored while being shaped are extracted, and information such as file managing information for accessing a filing system is extracted and accessed. Further, when writing data onto the disk 31, inside the format transformation program 24, a data stream shaped into the same format as the write of data received from a host 2 onto the disk 31 due to the storage device subsystem is prepared and written in the fixed length format.



LEGAL STATUS

[Date of request for examination] 28.03.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-184639

(43)公開日 平成11年(1999)7月9日

(51) Int.Cl.⁶
G 0 6 F 3/06

識別記号
3 0 1

F I
G 0 6 F 3/06

3 0 1 X
3 0 1 Z

審査請求 未請求 請求項の数6 OL (全9頁)

(21)出願番号 特願平9-350510

(22)出願日 平成9年(1997)12月19日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 北村 学

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 山本 彰

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 本間 繁雄

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージシステム事業部内

(74)代理人 弁理士 小川 勝男

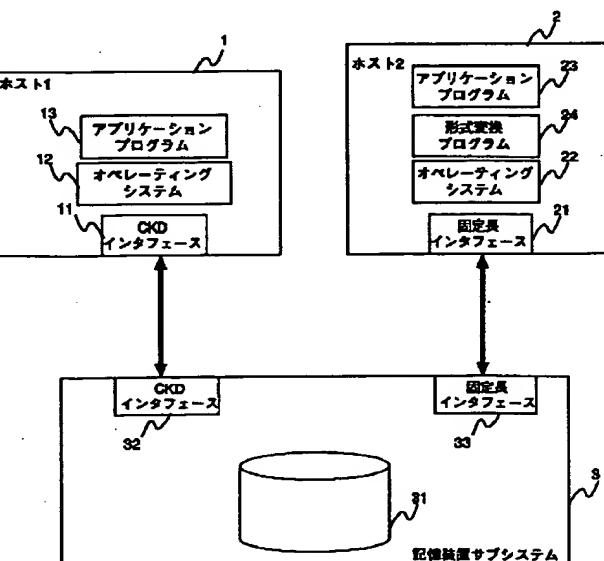
(54)【発明の名称】 計算機システム

(57)【要約】

【課題】記憶装置に格納するデータ形式が異なるシステムの間で、データの共有を可能にする。

【解決手段】ディスク31は固定長形式に従うディスク装置で、記憶装置サブシステム3はホスト1から来たCKD形式のレコードを固定長単位に整形して格納する。形式変換プログラム24ではディスク31の固定長ブロックを読み出して、整形して格納されているCKD形式のデータを抽出し、ファイル管理情報など、ファイルシステムにアクセスするための情報を取り出し、アクセスする。さらに、ディスク31へのデータ書き込み時は、形式変換プログラム24内部で、記憶装置サブシステムがホスト2から受けとったデータをディスク31に書き込むのと同じ形式に整形したデータ列を作成して、固定長形式で書き込む。

図1



【特許請求の範囲】

【請求項1】第1の計算機と、第2の計算機と、前記複数の計算機に接続する複数のインターフェースを有する記憶装置サブシステムとで構成される計算機システムで、前記第1の計算機は前記記憶装置サブシステムへのアクセスをカウントキーデータ形式に従って行い、前記第2の計算機は固定長ブロック形式で前記記憶装置サブシステムにアクセスする計算機システムにおいて、前記第1の計算機が前記記憶装置サブシステム内に作成したカウントキーデータ形式のレコードで構成されるファイルに、前記第2の計算機から固定長ブロック形式に従うインターフェースを介して、カウントキーデータ形式のレコードの読み込みを行う手段と書き込みを行う手段とを有することを特徴とする計算機システム。

【請求項2】請求項1に記載の計算機システムにおいて、前記書き込み手段は、前記第1の計算機が前記記憶装置サブシステム内に作成したファイルの管理情報を参照し、そのファイルのデータ形式に従ったレコードを書き込むことを特徴とする計算機システム。

【請求項3】請求項1に記載の計算機システムにおいて、該書き込み手段は、ユーザデータを書き換えるとともに、前記第1の計算機が前記記憶装置サブシステムに書き込んだファイルの管理情報を書き換えることを特徴とする計算機システム。

【請求項4】請求項1に記載の計算機システムにおいて、前記記憶装置サブシステムはカウントキーデータ形式のシリンド番号、ヘッド番号とで指定されるアドレスと、固定長ブロック形式の論理ブロック番号で指定されるアドレスとが1対1で対応し、前記書き込み手段は、書き込み要求のあったデータをカウントキーデータ形式のレコードとして扱い、該レコードの書き込み位置をカウントキーデータ形式のシリンド番号、ヘッド番号、レコード番号で認識しており、書き込み位置からカウントキーデータ形式レコードのカウント部を作成してカウントキーデータ形式のレコードに変換するとともに、該シリンド番号、ヘッド番号から、対応する固定長形式の論理ブロック番号を算出する手段を有することを特徴とする計算機システム。

【請求項5】請求項1に記載の計算機システムで、前記記憶装置サブシステムは、固定長ブロック形式に従う記憶方式の記憶装置と、前記第1の計算機がカウントキーデータ形式インターフェースを介して書き込んだレコードを固定長ブロック形式のデータ列に変換して前記記憶装置に書き込む手段を有する記憶装置サブシステムにおいて、前記第2の計算機の書き込み対象のレコードを固定長ブロック形式のデータ列に変換する手段は、前記記憶装置サブシステムの有する書き込み手段と同様の手段であることを特徴とする計算機システム。

【請求項6】請求項5に記載の計算機システムにおいて、前記書き込み手段は、前記カウントキーデータ形式

のレコードを前記固定長ブロックの大きさに分割して書き込みを行い、固定長ブロックの大きさに満たないデータの書き込みの際にはデータの直後にあらかじめ定められたデータ列を付加し、固定長ブロックの大きさにしてデータを書き込むことを特徴とする計算機システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理システムなどにおける記憶装置システムに係り、特に、複数のデータ形式に従うインターフェースを有する記憶装置サブシステムを備えた計算機システムに関する。

【0002】

【従来の技術】近年、パソコンやワークステーションなどの小型計算機を使用して、従来メインフレームで行われてきた業務を行う「ダウンサイジング」が盛んに行われている。メインフレームには、従来から行われてきた業務により大量の情報が蓄積されており、メインフレームで蓄積された情報を小型コンピュータからもアクセスしたいという要求がある。従来、異なるオペレーティングシステムの間でのファイルの共用あるいはファイル変換に関して、例えばワークステーション上のオペレーティングシステムから、パーソナルコンピュータ用のオペレーティングシステムで作成したファイルシステムのファイルをアクセスするための技術が知られている。パーソナルコンピュータのファイルシステムは、ファイルのフォーマットやファイル管理情報の位置などの情報が一般に公開されているものがあり、この技術では、ワークステーション上でパーソナルコンピュータのファイルフォーマットを解釈して、ファイルをアクセスする処理をワークステーションのオペレーティングシステムの機能に取り込んでいる。

【0003】メインフレームのファイルシステムにも、ファイル形式が一般に公開されているものがあるので、従来の技術のようにメインフレームのファイルフォーマットをオープンシステム上で解釈させれば、ファイル共用が可能だが、メインフレームは伝統的にディスクアクセスをカウントキーデータ形式（以下CKD形式）にしたがって行ってきたのに対し、オープンシステムはディスクアクセスを固定長データ形式（以下FBA形式）で行う。このため、メインフレームとオープンシステムでのファイル共有実現を行うにはデータ形式の違いを吸収する必要がある。特開平09-258908では、このデータ形式の違いを吸収する技術を示しており、それでは、記憶装置に書き込まれたカウントキーデータ形式のレコードをオープンシステム側が読み出すと、特定の形式に整形された固定長形式に見せる仕組みを有する記憶装置と、オープンシステム側には、その特定の形式に整形された固定長形式データを解釈してデータ部分を取り出す手段を用意することで、メインフレームが記憶装置に書き込んだデータセットをオープンシステムから読み出すこと

ができる。

【0004】

【発明が解決しようとする課題】特開平09-258908 ではオープンシステムからメインフレームのデータセットを読み出す方法については示されているが、メインフレームのファイルシステムに対してライトする方法については示されていない。オープンシステム側ではCKD形式データにあるカウント部の概念はなく、さらにその中に記述されるシリンド、ヘッドの概念もない。そのためオープンシステム側からCKD形式のデータを書き込むには、オープンシステム側でカウント部を作成する手段が必要となる。さらに、オープンシステム側で書き込んだデータセットをメインフレーム側で読み出せるために、メインフレーム側の作成したファイルシステムの管理情報などを更新する必要もあり、特開平09-258908 の方法では対応できない。

【0005】本発明の目的は、ディスク装置に、CKD形式に従うインターフェースと、FBA形式に従うインターフェースの両方を持たせ、オープンシステムからメインフレームのファイルシステム内データの書き込みを行わることにある。

【0006】

【課題を解決するための手段】上記課題を解決するための計算機システムの構成を以下に示す。本発明における計算機システムは、第1の計算機と、第2の計算機と、前記複数の計算機に接続する複数のインターフェースを有する記憶装置サブシステムとで構成される。第1の計算機は記憶装置サブシステムへのアクセスをカウントキーデータ形式に従って行い、第2の計算機は固定長ブロック形式で記憶装置サブシステムにアクセスする。第2の計算機は、第1の計算機が前記記憶装置サブシステム内に書き込んだカウントキーデータ形式のレコードを、固定長ブロック形式に従うインターフェースを介してアクセスする手段と、前記手段で読み出したレコードからユーザデータだけを取り出す手段と、第1の計算機が記憶装置サブシステム内に作成したレコードに書かれた、ファイルシステムなどの管理情報を解釈し、利用する手段を持つ。さらに、前記第2の計算機は、ライトデータをカウントキーデータ形式のレコードに変換し、固定長ブロック形式に従うインターフェースから記憶装置サブシステムへ書き込む手段をもつ。

【0007】また、第2の計算機の書き込み手段は、データ書き込みの際に、第1の計算機が前記記憶装置サブシステムに書き込んだファイルの管理情報を参照して、そのファイルのデータ形式を解釈し、そのファイルのデータ形式に従ったレコードを書き込む。また、第2の計算機の書き込みの完了とともに、このファイルの管理情報を書き換えることにより、第2の計算機が書き込んだレコードを第1の計算機から読み出すことを可能にする。

【0008】また、本発明における計算機システムの記憶装置サブシステムはカウントキーデータ形式のシリンド番号、ヘッド番号とで指定されるアドレスと、固定長ブロック形式の論理ブロック番号で指定されるアドレスとが1対1で対応しており、第2の計算機の書き込み手段は、書き込み要求のあったデータをカウントキーデータ形式のレコードとして扱い、書き込み位置のアドレスを用いてカウントキーデータ形式レコードのカウント部を作成してカウントキーデータ形式のレコードに変換し、記憶装置サブシステムへの書き込みはシリンド番号、ヘッド番号から、対応する固定長形式の論理ブロック番号を算出して、論理ブロック番号で記憶装置サブシステムにアクセスする。

【0009】また、本発明における計算機システムの記憶装置サブシステムは、固定長ブロック形式に従う記憶方式の記憶装置を有し、第1の計算機がカウントキーデータ形式インターフェースを介して書き込んだレコードを、定められた形式で固定長ブロック形式のデータ列に変換して記憶装置に書き込む手段を有する。また、第2の計算機の書き込み手段は記憶装置サブシステムがレコードを固定長ブロック形式のデータ列に変換する手段と同様の手段を有し、変換した固定長ブロック形式のデータ列を固定長ブロック形式に従うインターフェースを介して記憶装置サブシステムに書き込む。

【0010】また、上記書き込み手段は、カウントキーデータ形式のレコードを固定長ブロックの大きさに分割して書き込みを行う。レコードの終端が固定長ブロックの大きさに満たないデータになった場合には、データの直後にあらかじめ定められたデータ列を付加することで固定長ブロックの大きさに矯正し、記憶装置サブシステムに書き込む。

【0011】

【発明の実施の形態】図1に本発明を適用した計算機システムの構成例を示す。計算機システムは、ホスト1、ホスト2、そしてホスト1とホスト2に接続される記憶装置サブシステム3から構成される。

【0012】ホスト1は、記憶装置サブシステム3との間のインターフェースとしてカウントキーデータ形式（以下、CKD形式）に従うCKDインターフェース11を有し、オペレーティングシステム12によって制御される。さらに、アプリケーションプログラム13が存在し、オペレーティングシステム12上で動作する。メモリ、CPUなど計算機において必ず存在する構成要素も存在するが、実施形態の説明において、あまり重要ではないため、ここでは明記しない。

【0013】一方ホスト2は記憶装置サブシステム3と接続するためのインターフェースとして固定長ブロック形式に従う固定長インターフェース21を有し、オペレーティングシステム22によって制御される。さらにオペレーティングシステム22上で動作する、アプリケーショ

ンプログラム23と形式変換プログラム24が存在する。

【0014】記憶装置サブシステム3は、ディスク31、ホスト1と接続するCKDインターフェース32、ホスト2と接続する固定長インターフェース33から構成される。

【0015】固定長形式では、各データはブロックと呼ばれる領域に格納される。各ブロックは例えば512バイトなどの固定長である。各ブロックにはブロック番号(以下LBA)を指定してアクセスする。

【0016】一方CKD形式では、シリンド番号(C)、ヘッド番号(HH)、レコード番号(R)を指定してレコードにアクセスする。アクセスの最小単位はレコードである。以下、シリンド番号、ヘッド番号、レコード番号で表されるレコードアドレスをCCHHR、シリンド番号、ヘッド番号で表されるトラックのアドレスをCCHHと呼ぶ。CKD形式は、カウント部(以下C部)、キ一部(以下K部)、データ部(以下D部)で1つのレコードを形成する。C部にはCCHHRと、K部、D部の長さが入り、常に固定長である。K、D部は可変長で、C部に記された長さを有する。レコードが複数集まって、1つのトラックを形成するが、トラックの長さは固定長のため、各レコードの長さが異なる場合、1トラックに入るレコード数はそれぞれのトラックによって異なる。各トラックに入っているレコード数の情報やレコードの位置は管理されておらず、トラックの途中にあるレコードにアクセスしたい場合にはトラックの先頭レコードから順にカウント部を読み出し、次レコードの位置を知る必要がある。

【0017】また、トラックが複数集まって、1つのシリンドを形成する。1つのシリンドあたりのトラック数は一定である。

【0018】図2Aは、一般的なCKDディスク装置におけるトラック上のデータ配置を、図2BはCKD形式のデータを本実施形態の記憶装置サブシステム3に格納する場合のデータ形式を示す。ディスク31は固定長データ形式に従うものを使用し、CKD形式のデータを固定長データ形式に変換して格納している。

【0019】HA(51)はホームアドレスで、トラックの状態などを示す。R0C(52)はレコード0のカウント部、R0D(53)はレコード0のデータ部で、ユーザデータを格納することはできない。R1C(54)、R1K(55)、R1D(56)はそれぞれレコード1のカウント部、キ一部、データ部である。それぞれのフィールドの間にはギャップと呼ばれる一定の長さ

$$LBA = (C \times 1 \text{シリンド内トラック数} + H) \times 1 \text{トラック内ブロック}$$

数...

(式1)

で算出できる。

【0023】図3は、ホスト1がディスク31上に作成

を持った、データの格納されない領域が存在し、それぞれの区切りとなっている。また、C部、K部、D部の終端にはECCコード(60)(誤り訂正符号)が付加されている。C部の内容、ECCコード、ギャップはディスク装置内部で作られる。

【0020】記憶装置サブシステム3では、全トラックのHA(51)、R0C(52)、R0D(53)を別領域にまとめて格納し、固定長インターフェース33を介しては、HA(51)、R0C(52)、R0D(53)は見えないようにしてある。固定長インターフェース33からLBA0(61)にアクセスすると、シリンド番号0、ヘッド番号0、レコード番号1の先頭にアクセスすることになる。図2のR1Dのようにレコードが1ブロックを越える大きさのときは、次のブロックであるLBA1(62)に連続して書き込まれる。また、C、K、D部間のレコード内のギャップは存在しない。レコードの大きさは必ずしも固定長ブロックの境界と一致しないため、レコードの終端から、ブロックの終端までは空白(65)を入れ、0を並べてギャップの代わりとする。そしてレコード2は次のブロックであるLBA2

(63)から始まる。ホスト1が記憶装置サブシステム3に対してレコード書き込みを行う場合、CKDインターフェース32でC部、CKD部の終端に付加されるECCコードを作成し、固定長ブロックの大きさになるように空白65を追加してディスク31へ書き込みを行う。空白の内容は、何でも構わないが、本実施例では0を入れることとする。読み出しの際には、CKDインターフェース32が固定長のブロックを読み出してK部、D部を抽出してホスト1に渡すので、ホスト1は書き込みデータの形式やディスク31のどのブロックにレコードが書き込まれているかなどは考慮しなくて良い。一方ホスト2が記憶装置サブシステム3から読み出しを行う場合、例えば図2BのLBA0を読み込むと、ブロックの大きさでしか読み出せないため、LBA0の内容すべて、すなわちC部やECCを含んだ内容がホスト2に転送される。書き込みの場合もホスト2から転送したデータがそのままLBA0に書き込まれる。

【0021】トラック長はブロック長の整数倍となっており、すべてのトラックが同数のブロックで構成される。トラック内レコードの終端が、トラックの終端に一致しない場合は、トラックの終端まで0を入れて空白をつくり、全トラックが同じブロック数を占めるようにする。さらに各シリンドも同数のトラックで構成される。

【0022】そのため、CCHHR=(C, H, 1)の先頭にアクセスするには、LBAは

したファイルシステムの情報とファイル(データセット)を示す。ディスク31上の特定位置にはボリューム

ラベル71があり、ボリュームラベル71には、VTOC72 (Volume Table Of Contents) の書かれているアドレスがCCHHRで記されている。VTOC72にはそのディスク31に書かれているデータセットの管理情報が全て記録されており、例えばキー部44バイト、データ部96バイトのレコードの集まりになっている。キー部にデータセット名が記され、データ部にはそのデータセットの格納される位置情報などが記録される。位置情報として、レコード長、エクステントのある位置、エクステント内に書かれている最終レコードの位置を記録している。このデータセット1つ分の情報をDSCBと呼ぶ。

【0024】データセットはエクステントと呼ばれる領域に置かれる。エクステントは複数トラックが集まった連続領域で、1つのデータセットにつき、1つの場合もあれば、2つ以上存在することもある。エクステント複数ある場合、各エクステントが連続した領域に存在している必要はない。

【0025】ホスト1のアプリケーションプログラム13がデータセットを読み出す場合、VTOC72内の、該当するデータセット名の位置情報から、アクセスの開始位置、最終レコードの位置を取り出し、それによって読み出し位置のCCHHRを決定して読み込みを行う。これらの処理はオペレーティングシステム12が行う。また、ディスク31内のデータにはC部や、ECCなどのホスト1にとっては不要な情報が含まれるが、それらの部分はCKDインターフェース11、CKDインターフェース32で取り除かれ、ホスト1のオペレーティングシステム12には、K部、D部の実データだけが渡され、余計な情報は渡されない。

【0026】一方、ホスト2のオペレーティングシステム22ではホスト1のファイルシステムの構造を知らないだけでなく、CKD形式のデータがディスク31にどのように記憶されているかも知らないため、ホスト2のファイルシステムにアクセスするのと同じ方法ではアクセスできない。そこで、形式変換プログラム24を用意し、ホスト2のアプリケーションプログラム23は形式変換プログラム24を用いてホスト1のデータセットにアクセスする。形式変換プログラム24は、記憶装置サブシステム3がCKD形式のデータをディスク31に書き込む、またはディスク31に書き込まれたCKD形式のデータを取り出す方法と同じ方法を持っており、記憶装置サブシステム3からブロック単位にデータを読み出し、図2Bのような形式のデータから実データ部分を取り出す仕組みと、アプリケーションプログラム23から渡されたデータをレコードとみなして、CKDレコードに変換し、さらにそのCKDレコードをFBA形式に変換してディスク31に書き込む仕組みとを有する。さらに、VTOC72に書かれているデータセットの管理情報を解釈して、データセットの書かれている位置を知る

仕組みを持つ。以下、形式変換プログラム24がデータセットへの書き込みを行う時の流れを、オープン処理、ライト処理、クローズ処理の3つに分けて、フローチャートにて説明する。

【0027】図4は、形式変換プログラム24がデータセットのオープンを行う処理のフローチャートである。

【0028】形式変換プログラム24は1トラック分のデータを保持するトラックバッファと、DSCB1個分に相当するブロックのデータを保持するブロックバッファ、アクセス対象となるデータセットのトラック番号とレコード番号の組（これをレコードポインタと呼ぶ）、レコードポインタの示すレコード番号の、トラックバッファ内における位置を示すポインタ（これをバッファポインタと呼ぶ）を保持する。オープン処理ではまず、ボリュームラベル71のおかれているトラックを読んでVTOCの位置を調べる。ボリュームラベル71は常に固定位置に存在し、形式変換プログラム24はボリュームラベル71のディスク31上の位置をあらかじめCCHHRで知っている。例えばシリンド番号0、ヘッド番号0、レコード番号3に存在するとすると、シリンド番号0、ヘッド番号0に相当するLBA、すなわちLBA番号0を式1から計算し（ステップ101）、その位置からブロックを1トラック分トラックバッファに読み込む（ステップ102）。次に、読み込んだブロックの先頭に位置するレコード1のC部から、レコード1のK、D部の長さを調べることにより、レコード2の格納されているブロック位置がわかる（ステップ103）。後は再び、レコード2の先頭を読んでC部から、K、D部の長さを得て、レコード3のブロック位置を調べる（ステップ104）。これによりレコード3のD部がわかる。

【0029】続いてレコード3のD部からVTOC72の格納位置を読み出し（ステップ105）、その位置に相当するLBAを式1から計算し（ステップ106）、そのLBAから1トラック分のブロックをトラックバッファに読み込む（ステップ107）。続いてトラックバッファ内からオープン対象のデータセット名を検索する。カウンタrを用意し、rの初期値を1とする（ステップ108）。読み込んだトラックのレコードrのキー部を調べ、対象データセットの名前と一致するか判定する（ステップ109）。該当するレコードが見つかったら、D部から、データセットの格納位置を得て、このレコードのC、K、D部すべてを内部バッファに記憶しておく（ステップ110）。見つからない場合にはrを1増加し（ステップ115）、ステップ109に戻り次のレコードを検索する。また、読み出したトラックの終端まで調べても見つからない時には、次のトラックをディスク31から読み出し、レコード1から検索していく（ステップ113、114）。

【0030】次に、図5でライト処理を説明する。形式変換プログラム24はアプリケーションプログラム23

からレコード長と、データの内容を受けとり、レコード単位にライト処理を実行する。ただしディスク31への書き出しはブロックの整数倍単位で行う必要があるため、本ライト処理では、ディスク31への書き出しをブロックの整数倍の大きさである1トラック単位に行う。そのため、過去にライト要求があった場合、それまでのライト要求で受けとったデータをトラックバッファに記憶している。ライト処理ではまず、今回受け取ったデータがトラックバッファに書き込める大きさかどうかを判定して（ステップ201）、書き込めない場合には、現在のトラックバッファの書き込み位置からバッファ終端までの内容を0クリアし（ステップ202）、トラックバッファを記憶装置サブシステム3に書き込み（ステップ203）、バッファポインタをトラックバッファ先頭に戻し（ステップ204）、レコードポインタを次トラックのレコード1に移動する（ステップ205）。ディスク31にはトラックバッファに作られた内容そのままが書き込まれる。

【0031】ステップ206以降では、トラックバッファにCKD形式のデータを書き出す処理を行う。まず、レコードポインタとレコード長からC部の情報を作成して、トラックバッファのバッファポインタの指す位置に書き込み（ステップ206）、C部の後に渡されたデータを書き込む（ステップ207）。次のレコードは、次のブロックの先頭から書き込み始めるため、データがブロックの終端まで達していない場合には次のブロックの先頭まで0を入れる（ステップ208）。最後にバッファポインタを次のブロックの先頭に移し（ステップ209）、レコードポインタを次レコードに移す（ステップ210）。

【0032】次に図6で形式変換プログラム24のクローズ処理を説明する。まず、直前までのライト処理で、トラックバッファにデータが書き込まれているかチェックし（ステップ301）、データが書き込まれている場合には、現在のバッファポインタの示す位置からトラックバッファの終端まで内容を0クリアし（ステップ302）、ディスクに書き出す（ステップ303）。続い

て、VTOC72内の最終レコード位置を書き換えるため、記憶しておいたブロックバッファの内容のうち、最終レコード位置の情報を書き換え（ステップ304）、ブロックバッファをディスクに書き出す（ステップ305）。これでクローズ処理が完了する。

【0033】

【発明の効果】本発明の計算機システムによれば、CKD形式とFBA形式という異なったデータ形式の記憶方式に従うシステム同士が、データを共有することができる。これにより、データ交換をより柔軟で、高性能に行うことが可能となる。

【図面の簡単な説明】

【図1】本発明を適用したディスク記憶システムの構成図である。

【図2】記憶装置サブシステムにCKD形式データの格納される様子を示す図である。

【図3】ディスク装置内の、データセットの格納形式を示す図である。

【図4】データセットのオープン処理のフローチャートである。

【図5】データセットのライト処理のフローチャートである。

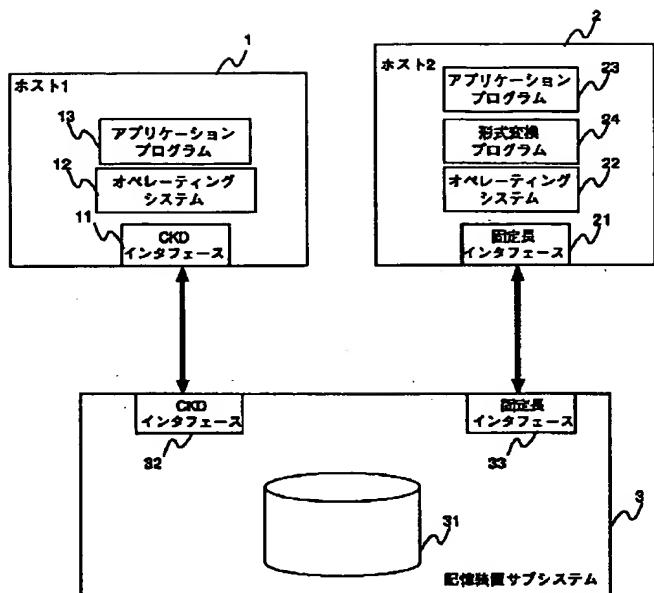
【図6】データセットのクローズ処理のフローチャートである。

【符号の説明】

1：ホスト、2：ホスト、3：記憶装置サブシステム、11：CKDインターフェース、12：オペレーティングシステム、13：アプリケーションプログラム、21：固定長インターフェース、22：オペレーティングシステム、23：アプリケーションプログラム、24：形式変換プログラム、31：ディスク、32：CKDインターフェース、33：固定長インターフェース、51：HA、52：R0C、53：R0D、54：R1C、55：R1K、56：R1D、57：R2C、58：R2K、59：R2D、60：ECCコード、61：LBA0、62：LBA1、63：LBA2、64：LBA3、65：空白、71：ボリュームラベル、72：VTOC。

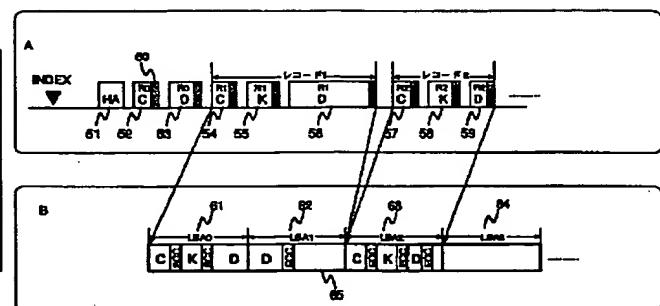
【図1】

図1



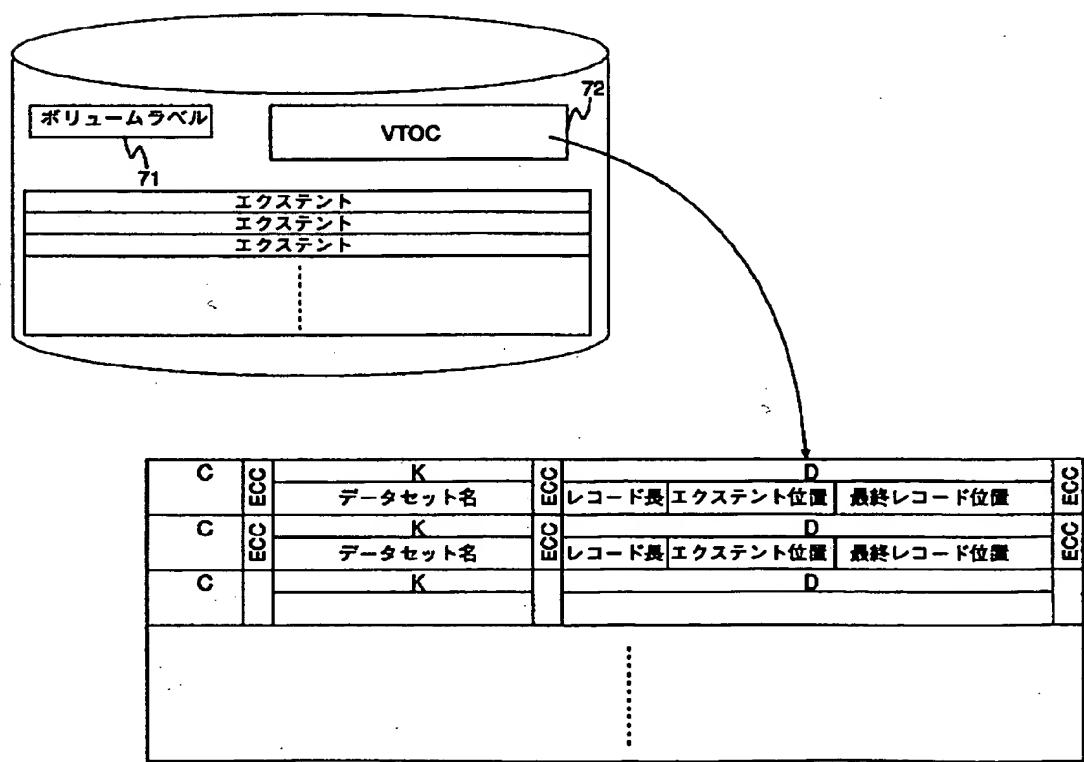
【図2】

図2



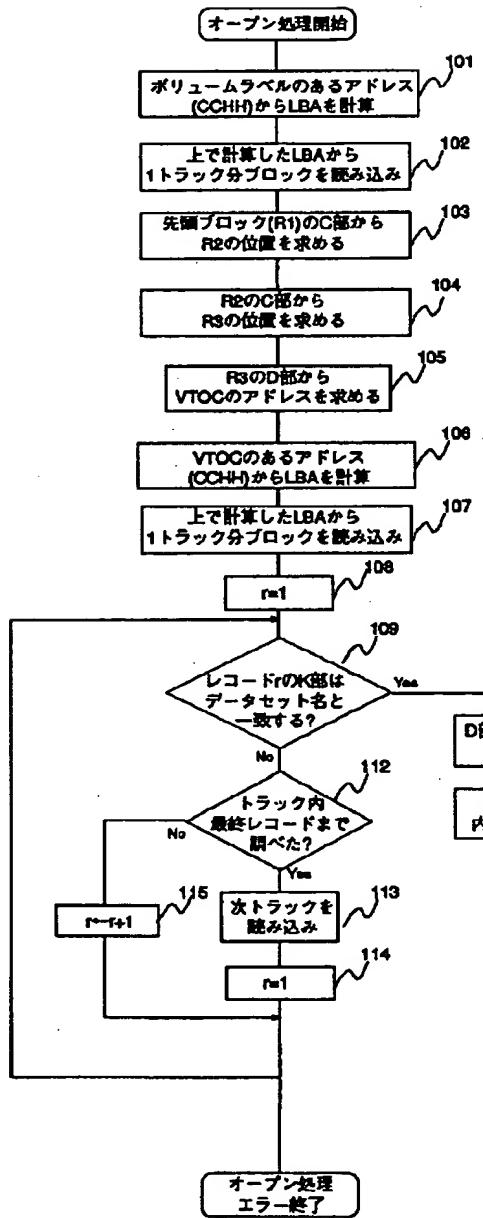
【図3】

図3



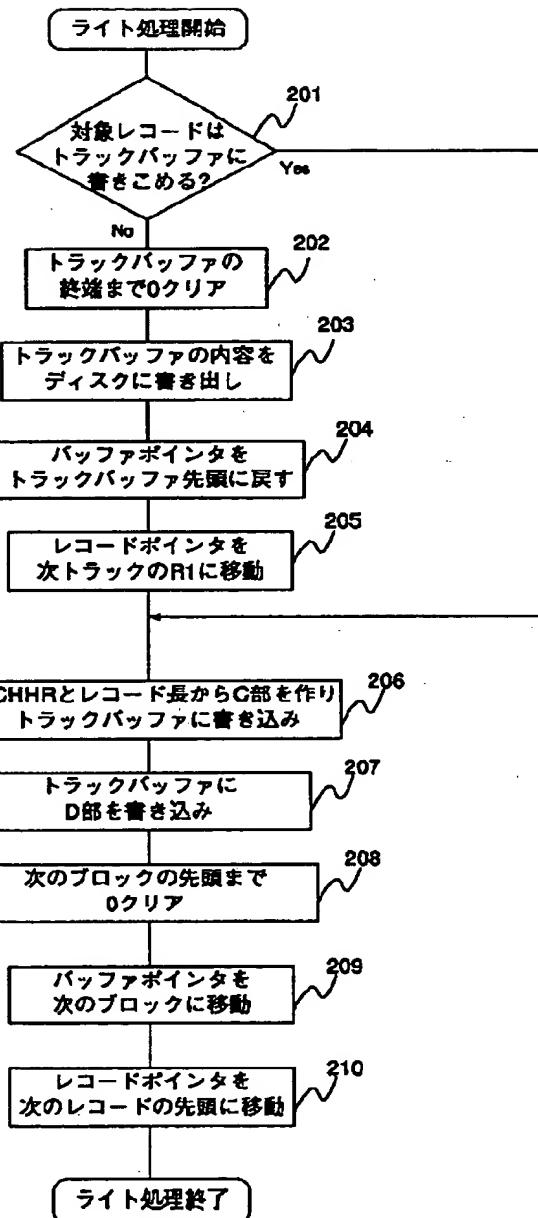
【図4】

図4



【図5】

図5



【図6】

図6

